



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/757,718	01/13/2004	John Randall West	07464-0004	3834
25213	7590	10/06/2006	EXAMINER PHAM, THAI V	
HELLER EHRMAN LLP 275 MIDDLEFIELD ROAD MENLO PARK, CA 94025-3506			ART UNIT 2191	PAPER NUMBER

DATE MAILED: 10/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

10/757,718

Applicant(s)

WEST ET AL.

Examiner

Thai Van Pham

Art Unit

2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 13 January 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1 - 55 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-55 is/are rejected.
- 7) ☒ Claim(s) 55 is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 13 January 2006 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date 5/18/2006.
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_.

### DETAILED ACTION

This is the initial office action based on the application filed on September 29, 2006.

Claims 1 – 55 are currently pending and have been considered below.

#### ***Specification***

1. The disclosure is objected to because of the following informalities:

-- Paragraph [0032] in the specification refers to "*flow control structure 32*" which is not identified anywhere in the drawings.

-- Paragraph [0039] in the specification refers to "*an internal identification field 56*" which is not identified anywhere in the drawings.

-- The last sentence of paragraph [0039] in the specification refers to "*the same application object 16*" which should correctly be identified as "... 24".

-- Paragraph [0042] in the specification use the same number "122" to identify both "*logic 122*" and "*an abstract test case representation 122*".

-- Paragraph [0042] in the specification use the same number "126" to identify both "*logic 126*" and "*an application metadata repository 126*".

-- Paragraphs [0056] and [0058] in the specification use the same number "222" to identify both "*data stores*" and "*input data*".

-- Paragraphs [0043] – [0055] are copies of paragraphs [[0030] – [0042], respectively.

Appropriate corrections are required.

***Drawing***

2. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they do not include the following reference sign(s) mentioned in the description: "32" and "56" (See objections to Specification above.) Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

3. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because reference character:

-- "54" has been used to identify both "*Application Meta Data Repository*" in Figure 4 and "*Application Event Effect*" in Figure 6.

-- "381" in Figure 11 is incorrectly labeled – it should be "318" as referred to in paragraph [0054] in the specification.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted

Art Unit: 2191

after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Appropriate corrections are required.

***Claim Objections***

4. Claim 55 is objected to because of the following informalities: containing 2 ending periods. Appropriate correction is required.

***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1, 5 – 7, 8 – 25 are rejected under 35 U.S.C. 102(e) as being anticipated by **Rosaria** (6,976,246).

-- Claim 1:

**Rosaria** discloses *a method for generating test cases, comprising:*

- *providing rule-based generation of test cases from an abstract representation that includes application states, external interaction sequences and input data of test cases from data stores (Fig. 3, page 6: lines 38 – 65; Fig. 6, page 9: line 16 – page 10: line 56; generating “test sequences” using “model editor” and “rules editor”);*
- *validating generated test cases (Figs. 3 and 10; page 14: line 62 – page 15: line 8; generating test sequences in accordance to test algorithm specified in “graph traversal program”); and*
- *converting the test cases to test scripts (Fig. 11, page 15: lines 9 – 21).*

-- Claim 5:

**Rosaria** discloses *the method of claim 1,*

- *wherein an application state represents a runtime snapshot of application under test which defines the context of external interaction (Fig. 3, page 6: line 66 – page 7: line 11; a state in the finite state model).*

-- Claim 6:

**Rosaria** discloses *the method of claim 5,*

- *wherein the application state includes a set of application objects, its attributes and attribute values (Fig. 3, page 6: lines 38 – 65; Fig. 7, page 10: line 57 – page 11: line 6; state attributes and rules setting).*

-- Claim 7:

**Rosaria** discloses *the method of claim 5,*

- *wherein the application states corresponding to a test case are arranged in a hierarchical manner (Figs. 2 and 9, page 13: line 31 – page 14: line 55; states and their dependencies).*

-- Claims 8 and 9:

**Rosaria** discloses *the method of claim 1,*

- *wherein the external interaction sequences represent events invoked by external agents on the application objects, and the external agents are human agents or other software agents (Fig. 3, page 6: lines 18 – 27; state transitional conditions implies that events that are defined using “model editor” and “rules editor” to cause a state transition are, in fact, interaction sequences that can be external to an application object, which are events triggered by external agents associated with the application itself).*

-- Claim 10:

**Rosaria** discloses *the method of claim 8*,

- *wherein the interaction sequencing includes flow control structures for capturing sequential, concurrent, looping and conditional interactions* (Fig. 8, page 11: line 7 – page 13: line 30; transitional operators and state flow control in “rules editor”).

-- Claim 11:

**Rosaria** discloses *the method of claim 1*,

- *wherein the validation of generated test cases includes internal* (Fig. 8, page 11: line 7 – page 13: line 30; “rules editor”) *and external validation* (Fig. 10, page 14: line 62 – page 15: line 8; “graph traversal algorithm”).

-- Claim 12:

**Rosaria** discloses *the method of claim 11*,

- *wherein the internal validation ensures that the components of the test case definition, external interaction sequences and input data are consistent with each other and with an application object model* (Fig. 8, page 11: line 7 – page 13: line 30; “rules editor”).

-- Claims 13 – 17:

**Rosaria** discloses *the method of claim 12*,

- *wherein an application object model is a metadata representation for modeling application under test* (Fig. 9, page 13: line 31 – page 14: line 55; data structure of states);



- *wherein the metadata representation includes object type definitions for application objects (Fig. 9, page 13: line 31 – page 14: line 55; data structure of states and their attributes);*
- *wherein the metadata representation includes attribute definitions for each application object type (Fig. 9, page 13: line 31 – page 14: line 55; data structure of states and their attributes);*
- *wherein the metadata representation includes definition of methods and events that are supported by each application object type (Figs. 2 and 8, page 11: line 7 – page 13: line 30; transitional conditions and events); and*
- *wherein the metadata representation includes definition of effects of events on an application state (Figs. 2 and 8, page 11: line 7 – page 13: line 30; “state transitions” caused by external/internal events).*

-- Claim 18:

**Rosaria** discloses *the method of claim 14,*

- *wherein application object type definitions include additional categorization of each application object types into hierarchical, container and simple types (Fig. 9, page 13: line 31 – page 14: line 55; states in state model).*

-- Claim 19:

**Rosaria** discloses *the method of claim 18,*

- *wherein the hierarchical object types are associated with an application state of its own, wherein application object types that can contain instances of other objects are termed container types (Figs. 2 and 9; independent and dependent states).*

-- Claims 20 and 21:

**Rosaria** discloses *the method of claim 19*

- *wherein the state associated with a hierarchical application object type is a modal application state or a non-modal application state; wherein a modal application state restricts possible interactions to application object instances available within the current application state (Fig. 2 and 9; types of “states” in a FSM, such as Mealy and Moore types).*

-- Claims 22 – 25:

**Rosaria** disclose *the method of claim 17,*

- *wherein the effects of events on an application state capture one or more consequences of the event to the application state (Figs. 2 and 8, page 11: line 7 – page 13: line 30; “state transitions” caused by external/internal events);*
- *wherein a consequence of an event is selected from, creation of a new object instance of a given type, deletion of an object instance of a given type, modification of attributes of an existing object instance and selection of an instance of an object type (Fig. 8, page 11: line 7 – page 13: line 30, “rules” defining program flow of “state model”);*

- *wherein creation of a new instance of an object of type that is hierarchical results in creation of a new application state* (Fig. 8, page 11: line 7 – page 13: line 30, dependent states); and
- *wherein selection of an object instance of type that is hierarchical results in selection of the application state associated with that object instance* (Fig. 8, page 11: line 7 – page 13: line 30; states and their dependencies as characterized according to the application structure and properties).

6. Claims 1, 11, 26 – 31, 33, 34, 37, 39 – 55 are rejected under 35 U.S.C. 102(e) as being anticipated by **Kossatchev** (6,698,012).

-- Claim 1:

**Kossatchev** discloses *a method for generating test cases, comprising:*

- *providing rule-based generation of test cases from an abstract representation that includes application states, external interaction sequences and input data of test cases from data stores* (Figs. 5 and 6; page 5: line 22 – page 7: line 16; conversion of RAISE Specifications and Script Driver skeletons into Formal Specification with Higher Abstraction Level);
- *validating generated test cases* (page 4: line 48 – page 5: line 10; basic driver generator and script driver generator); and
- *converting the test cases to test scripts* (Fig. 6; page 7: line 19 – page 11: line 45).

-- Claim 11:

**Kossatchev** discloses *the method of claim 1,*

Art Unit: 2191

- *wherein the validation of generated test cases includes internal (page 4: lines 48 – 63) and external validation (script driver generator; page 4: line 64 – page 5: line 10; basic driver generator).*

-- Claims 26 – 27:

**Kossatchev** discloses *the method of claim 11,*

- *wherein the external validation validates the generated test case against the application metadata repository; and wherein the application metadata repository contains definition of application objects and nature of their interactions within the application under test (Fig. 6, page 9: lines 11 – 65; “script driver skeletons”).*

-- Claims 28 - 30:

**Kossatchev** discloses *the method of claim 26,*

- *wherein the external validation serves as a static verification test for the test cases (page 8: line 48 – page 11: line 13; “script driver skeletons”);*
- *wherein the external validation increases productivity by pointing out invalid test cases; and wherein the external validation increases productivity by pointing out inconsistencies in statically verifiable application behaviors (script driver functionality; page 8: line 48 – page 11: line 13).*

-- Claim 31:

**Kossatchev** discloses *the method of claim 1,*

- *wherein the test scripts are test cases represented in a scripting language (Fig. 2, page 3: lines 14 – 63; “test suites” converted from “test cases”).*

-- Claims 33, 34, and 37:

**Kossatchev** discloses *the method of claim 1, further comprising:*

- *providing rules for selection of components of test case definition, external interaction sequences and input data (Figs. 3 and 5, page 5: line 64 – page 6: line 35; generating formal specification);*
- *rules for data driven test case generation; wherein the selection rules are specified using query languages; and wherein the query language is Application Programming Interface (API) called from code written in a programming language (page 1: lines 10 – 19; page 2: lines 44 – 65).*

-- Claim 39:

**Kossatchev** discloses *the method of claim 33,*

- *wherein the data driven test case generation involves composing the test case as dictated by the input data (Fig. 6, page 7: lines 31 – 35; RAISE specification).*

-- Claims 40 and 41:

**Kossatchev** discloses *the method of claim 39,*

- *wherein the availability of multiple datasets for the input data will result in generation of multiple test cases or external interaction sequences repeated within a loop control structure for each dataset (Fig. 6, page 8: line 48 – page 10: line 49; test sequence in serial and parallel compositions as well as iterations); and*

- *wherein the availability of multiple datasets for a portion of the input data will result in the interaction sequences corresponding to this portion of input data repeated within a loop control structure (Fig. 6, page 8: line 48 – page 10: line 49; test drivers).*

-- Claims 42 and 43:

**Kossatchev** discloses *the method of claim 39,*

- *wherein each element of input data can be flagged valid or invalid (Fig. 10, page 16: line 27 – page 17: line 43; specifying sequential or concurrent testing behaviors in test plan); and*
- *wherein the presence of validity flag in the input data that is different from the one corresponding the input data when the test cases was recorded or authored, results in the generator including appropriate interaction sequences for exception handling (Fig. 11, page 17: line 46 – page 19: line 4; handling concurrent procedure testing – serial procedure testing being the default).*

-- Claim 44:

**Kossatchev** discloses *the method of claim 1, further comprising:*

- *converting test case from internal representation to a scripting language through language mapping (Fig. 6; conversion of RAISE specifications and script driver skeletons to RSL basic drivers, RSL test case parameters, and RSL script drivers).*

-- Claims 45 – 48:

**Kossatchev** discloses *the method of claim 44,*

- *wherein the mapping is used to map external interactions captured as events on application object to appropriate statements in the scripting language environments (Fig. 6, page 11: lines 14 – 44; translation compilation of RSL “basic drivers”, RSL “script drivers”, and RSL “test case parameters” into target specific language);*
- *wherein more than one language mappings are provided at the same time (Fig. 6, page 11: lines 14 – 44; the language is target dependent);*
- *wherein the generated test case are converted to more than one scripting language at the same time environments (Fig. 6, page 11: lines 14 – 44; the drivers and parameters are in target specific language); and*
- *wherein generating test cases in multiple scripting language allows generation of test scripts for multiple test execution environments (Fig. 6, page 11: lines 14 – 44; translation compilation of RSL basic drivers, RSL script drivers, and RSL test case parameters into target specific language).*

-- Claim 49:

**Kossatchev** discloses *a computer system, comprising:*

- *a processor (an inherently essential hardware element in any computer system)*
- *a memory (an inherently essential hardware element in any computer system) coupled to the processor, the memory storing rule-based generation of test cases from an abstract representation that includes application states, external interaction sequences and input data of test cases from data stores to produce test cases (Figs. 5 and 6; page*

Art Unit: 2191

5: line 22 – page 7: line 16; conversion of RAISE Specifications and Script Driver skeletons into Formal Specification with Higher Abstraction Level);

- *logic for validating generated test cases* (page 4: line 48 – page 5: line 10; basic driver generator and script driver generator); and

- *logic for converting the test cases to test scripts* (Fig. 6; page 7: line 19 – page 11: line 45).

-- Claim 50:

**Kossatchev** discloses *the system of claim 49*,

- *wherein the logic that validates the test cases provides that the components of the test case definition, external interaction sequences and input data are consistent with each other and with an application object model* (Fig. 1, page 4: line 17 – page 5: line 10; “test driver generator”).

-- Claims 51 and 52:

**Kossatchev** discloses *the system of claim 49*,

- *wherein the logic that validates the test cases is external validation logic* (page 4: line 64 – page 5: line 10; “script driver generator”); and

- *wherein the external validation logic includes validating a generated test case against an application metadata repository* (Fig. 6, page 9: lines 11 – 65’ “script driver skeletons”).

-- Claims 53 and 54:

**Kossatchev** discloses *the system of claim 49, further comprising*:



- *logic for providing rules for selection of components of test case definition, external interaction sequences and input data; wherein and the rules are data driven test case generation (Figs. 3 and 5, page 5: line 64 – page 6: line 35; generating formal specification); and*
- *logic for providing data driven test case generation (Fig. 6, page 9: lines 11 – 65; “generating script driver skeletons”).*

-- Claim 55:

**Kossatchev** discloses *the system of claim 54,*

- *wherein the logic for providing data driven test case generation includes composing the test case as dictated by the input data (Fig. 6, page 7: lines 31 – 35’ “RAISE specification”).*

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 2 – 4 are rejected under 35 U.S.C. 103(a) as being obvious over **Rosaria** (6,976,246).

-- Claims 2 – 4:

**Rosaria** discloses *the method of claim 1* but does not explicitly disclose that

- *a data store is a relational database management system, an XML database management system, or a file system.*

Official Notice is taken that a file system, relational database management system, or XML database system had been among the well-known conventional models of database system in the art at the time the invention was made. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to implement the database system using a relational database management system or an XML database management system to provide the method with a standardized interface and protocol for storing data.

8. Claims 32, 35, 36, and 38 are rejected under 35 U.S.C. 103(a) as being obvious over **Kossatchev** (6,698,012).

-- Claim 32:

**Kossatchev** discloses *the method of claim 31* but does not explicitly disclose that

- *the scripting languages can be typed or un-typed programming languages used for recording or authoring test cases.*

Official Notice is taken that typed programming languages such as Pearl and Tcl as well as un-typed programming languages such as DOS and UNIX Shells commands had been well known and used in scripting at the time the invention was made. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made that to use one or a mixture of typed and un-typed languages for the scripting language to provide the user with ease and comfort in understanding and writing test scripts.

-- Claims 35, 36 and 38:

**Kossatchev** discloses *the method of claim 34* but does not explicitly disclose that

- *the query language is Structured Query Language (SQL) or XML Query (XQuery); and*
- *wherein the use of query languages allows test cases to be generated from live customer data.*

Official Notice is taken that SQL and XQuery had been well known and used as query languages at the time the invention was made. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use SQL or XQuery as the query language to provide the user with ease and comfort in

Art Unit: 2191

understanding and writing query commands in testing development. Furthermore, it would have been also obvious that *the use of query languages allows test cases to be generated from live customer data* because this is the main objective of using query languages.

***Conclusion***

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

-- **Jean S. Hartman et al.** (6,505,342), System and Method for Functional Testing of Distributed Component-Based Software. Siemens Corporate Research Inc..

-- **Parker et al.** (5,600,789), Automated GUI Interface Testing. Segue Software, Inc..

-- **Srivastava et al.** (6,609,248), Cross Module Representation of Heterogeneous Programs, Microsoft Corporation.

-- **Venter** (2004/0194072). Multi-Language Compilation.

Art Unit: 2191

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Thai Van Pham whose telephone number is (571) 270-1064. The examiner can normally be reached on Monday - Thursday, 9am - 5pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



TVP



Wei Y. Zhen  
Supervisory Patent Examiner